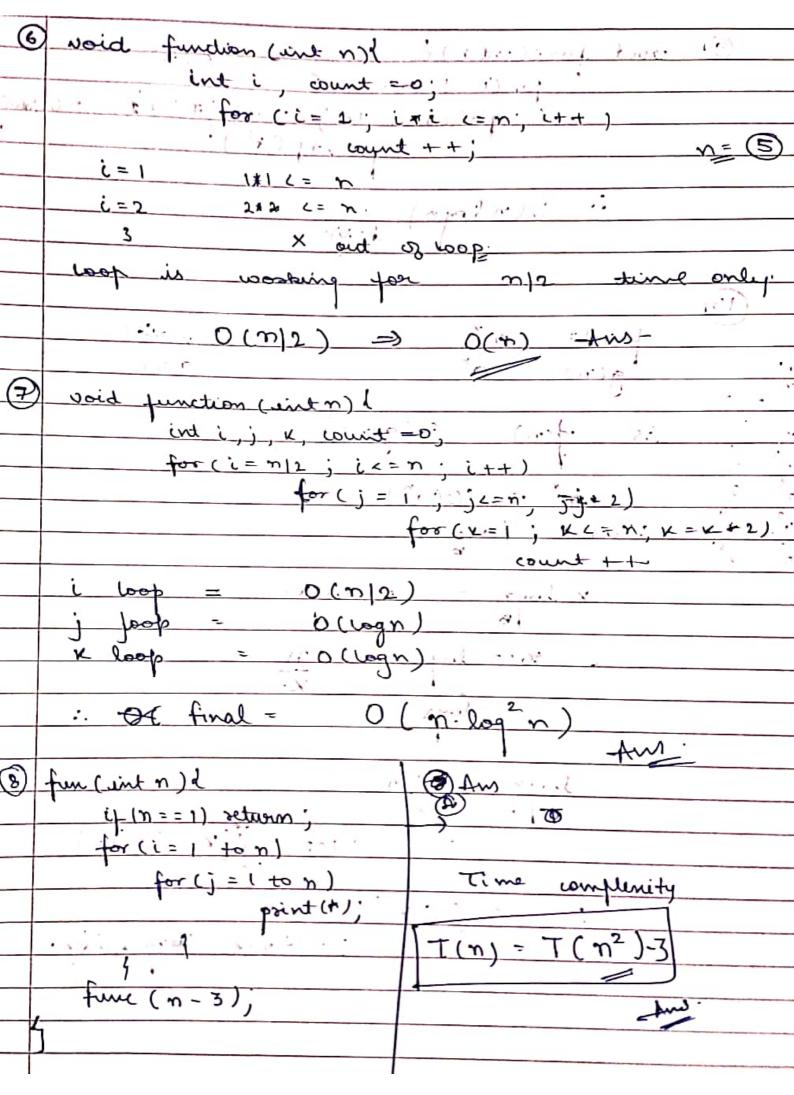
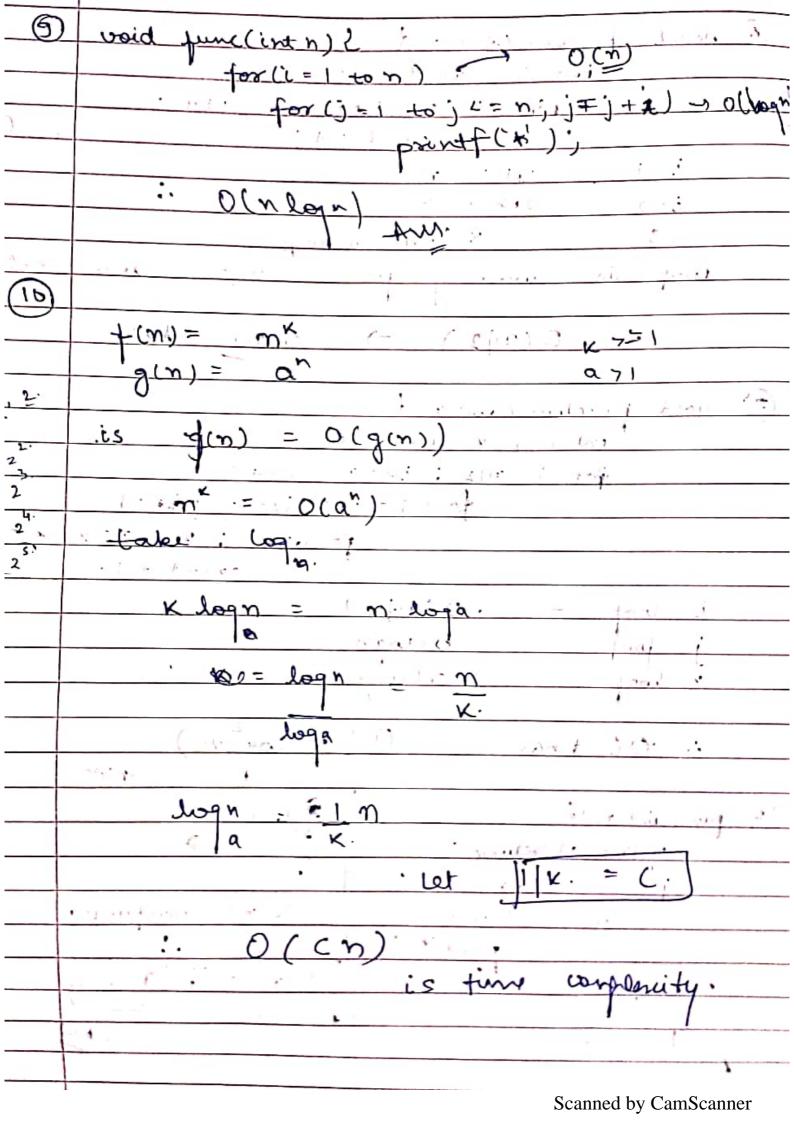
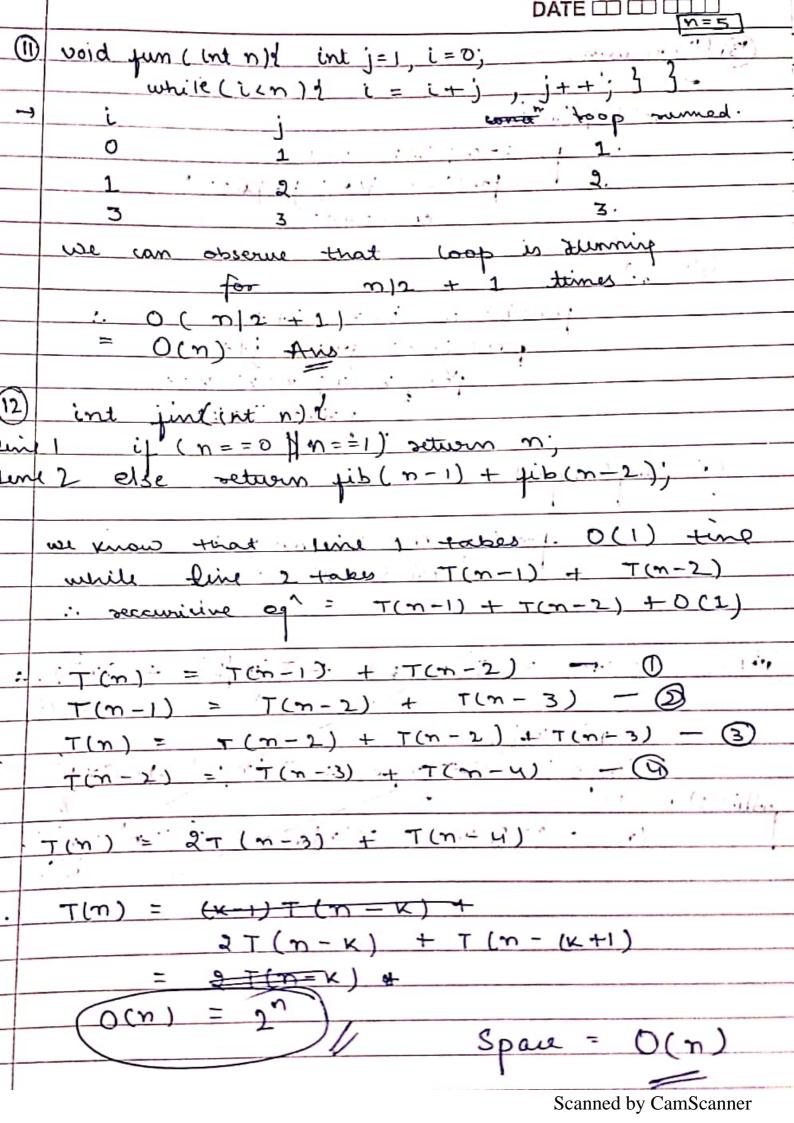
	Assignment #1
_C	
	Asymptotic nobations are methods languages using which
	we can define the sun-ring time of algorithm
	based on input size!
	To represent the inferi & love bounds; we meed
	some bind of syntax, & this is nepresented
	in form of function (con).
•	Logarithmic -> log n London -> n
•	Quadratic -> 2 Polynomial - m2
٠	Exponential - a
	(1. 21 - 1. 1) 7 2 . 1) 7
_2	
	for(i=1 to n) 1 i = i *2; 4,
	"i' is doubling grade everytime
	por K step -) 2x = n & for (x+1) we are
	taking larp both side.
	logs = logn.
	nuil .
_	K = log h; Tim Complexity - O (logn
(3)	T(n) = 3T(n-1) if in 70, otherwise 1.
	F(m): 3(3T(m=1))
	7
	T(n) = a T(n-b) + f(n) [Master Theorem]
	$\Delta = 3  b = 1$
	· ·
-	$\frac{(n)=0}{T(n)=0} \left( \frac{x=0}{n^k} \right)$
-	(na)
	$= T(m) = O(n^{6} O^{n})$
	Vardhman $T(n) = O(n^{\circ} a^{n})$ PAGE
-	



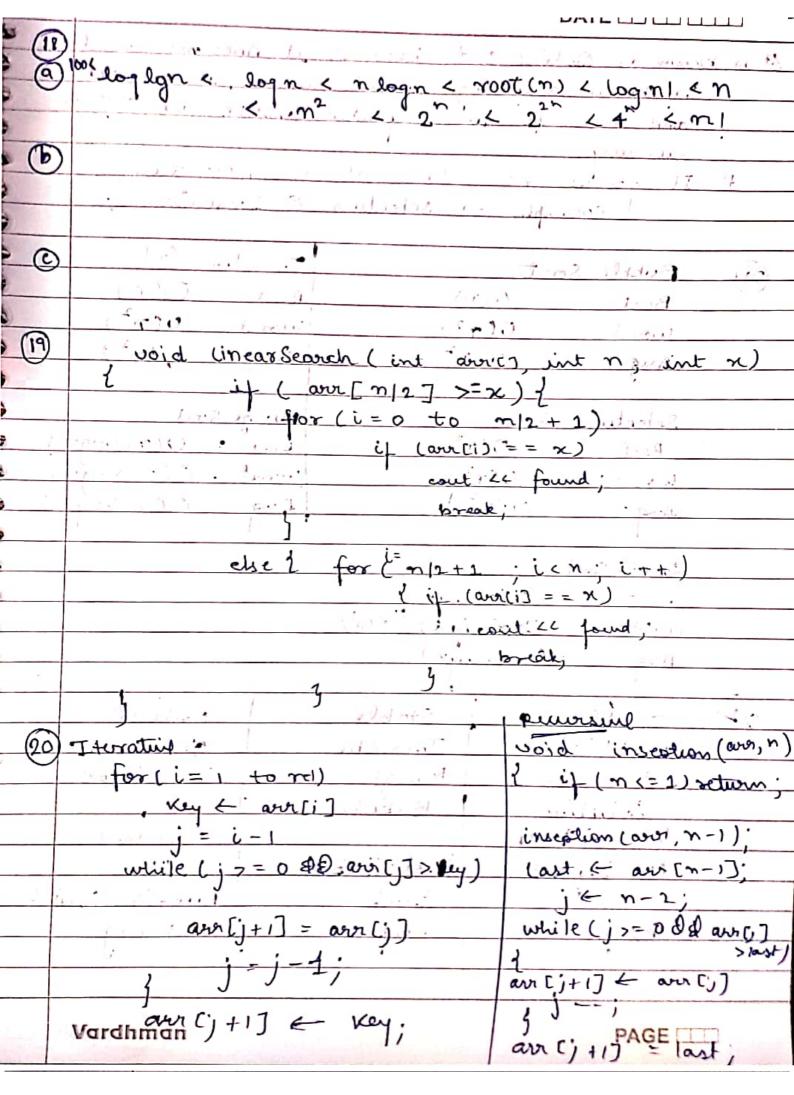




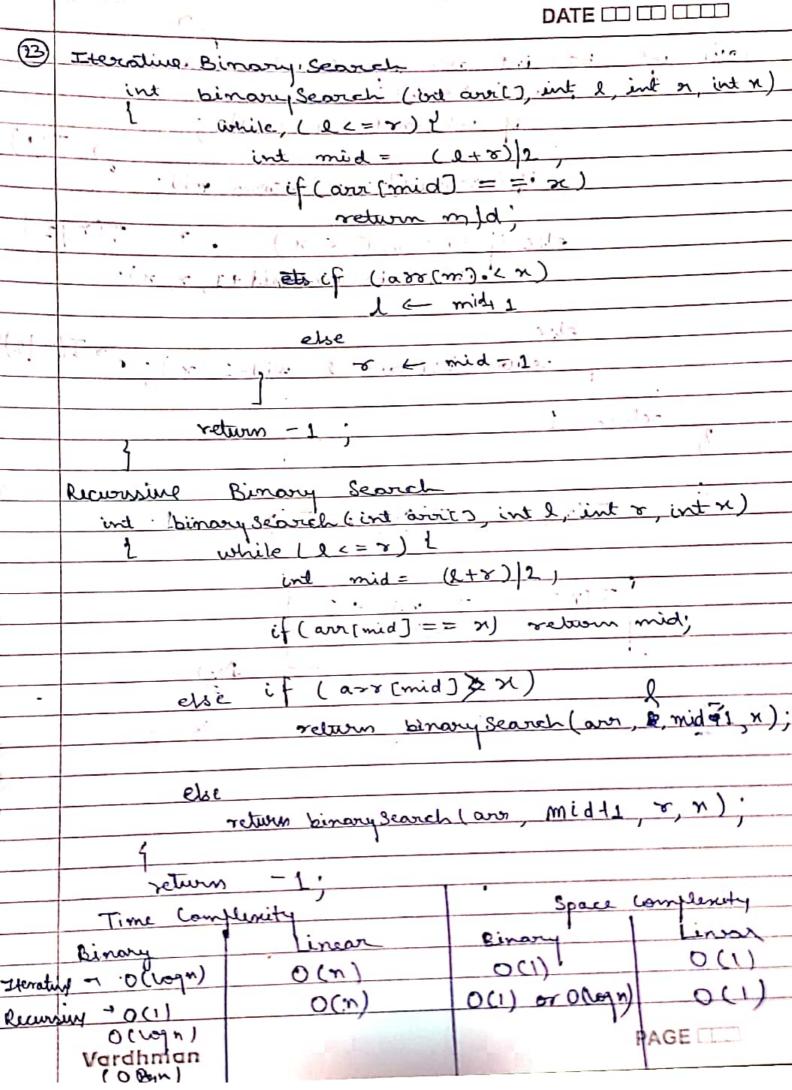
0 (	
(13) (1	(n logn)
1 -	void f (MD) ?
	ton (i=0; ikn; i++)
	for (j=0, j <n; j="j+2;)&lt;/td"></n;>
	count ++;
	Survey of the same
<b>③</b>	m <sup>3</sup>
	for (i = 0, i <n; i++)<="" th=""></n;>
	for ( j=0; j \ n; j++).
	for (K=0: KCN, K++)
	point ("onev");
	The second of th
3	· log (ling n) ! !
)	100 jor (i = 0 ", i < logn; i = i = 2). "
	- 1 1 - point ("ri").
,	

	DAIE
(4)	$T(n) = T(n 1) + T(n 2) + cn^{2}$
	using Master's Theorem
	we know that; T(n/2) >= T(n/4)
	· o T(n) <= 2 T(n/2) + cm2
	Apply Marter's theosen to RMS
	we feel to a feel and the
3	T(n) <= 0 (n2) · 1 · Since
	$T(n) = O(n^2)$ $T(n) = O(n^2)$
	Also (2).
_	T(n)1>=·cn².
	$T(m) = O(m^2) \qquad : T(m) = O(m^2)$
	T(m) = 2 (m2) - 1 (Am.
	. ==
_(15	
	int fun(intn)?
	for (i=1; i <= n; i++)
	For (j = 1; j < m; j = j + 1/9/2 O(1)
	- for the in loop = O(n)
The second	for the j loop = O(logn) williams
	·· O (n. logn) Au.
_	
(10	for (1=2; i <= n; i = pow(i, x))
	2/10(1)
	X=2.
	the transmit
	2 2 (2'
	4 . 2
	16 , X
	x hop hap (m)
	2 logen).
	Vardhman : O( wy(wq w)). PAGE
	oc william.

	DATE
(IP)	quick sout
	when quick part mostide divides the arrivary
	in the true is to a regard and 1-1.
	when quick sort repeatedly divides the array out of the the out of 99.1. and 1.1.
:- Recu	vence Relation:
	Let euvrent problem = n
	- Simily me have to call.
	1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1
1	-Quicksort Word (asl
	T(n) = T(n-1) + T(1) + Cn
(	The state of the s
Α,	T(n) = T(n-1) + O(n)
	<b>n</b>
	> 0(K)
	V=1
	= 0 (\(\sum_{\cute{\cie\cute{\cute{\cute{\cute{\cute{\cute{\cute{\cute{\cute{\cute{\cie\cute{\cute{\cute{\cute{\cute{\cute{\cute{\cute{\cute\}}}}}}}}}
17/10	the second second
	=
	O(n2) is the time compensity.
Recursion	m Ferra
	n -> 'n
	L S
	, n-1
_	1 : 1 = 1 n-2 = in-1 : in
A. J.	<u> </u>
H the p	irtioning 1 n-3 - n-2.
is ma	ximally unbalances
resis to	y temesing
step of	algorithm.
-	1 1 2
	-1/21
K	· O(n2)
	Vardhman



A+ :.	M	a . A . )				
It is known as Online Sort because it does not need !!!						
	ofina woner or	to senow anything about values, it will sort and the info is nequested while the olgo is				
	and me unto	and the info is requested WHILE The				
		running.				
	e sans ne	It grahs new value at every iteration.  example - selection & Insertion.				
	Skanples	-> serccuss_	( Distriction			
<u>(21)</u>	Bubble Sort		Insution Soit			
	Best = O(n)		Best = O(n)			
	Worst = 01		10 .A O(m-)			
i N	Aug . = , 0		Aug = 0 (m2)			
	-1.5					
	Selection: Sort.		Merige Sort			
	Best : - O(n2) . :		Best = O(nlogn)			
	Worst = 00		word: O(n logn)			
	Aug - 0(7)		Aug. : O(n lagn)			
	Quick Sort :.	1111 1 7 501	i water			
	Best i O'm					
	World To Olar	san) 1/0(n2)				
	Ang ~ O(n)	( N 00				
	71					
(22-)	In Place	Stable	1 online			
•	Bubble	Insertion	selection :			
n auto	Insertion	Mirge .	Insertion			
	Selection	Bubble	1. 1. 1.			
• •	Quick, Heap		affini			
- :	Not Infact.	"Not Stable	, V.V.,			
		Quick	Bubble, Orick			
	Merge	Hlas.	Merge			
. •.			1			
		•				



else reliun -1